# Implementing a Scripting Interface

## Session 112

These are confidential sessions—please refrain from filming or taking pictures.

Application Technologies

# Implementing a Scripting Interface

**John Comiskey**
AppleScript Engineer

http://developer.apple.com/wwdc2005/

# Assumptions

- You know Cocoa and Objective-C
- Your application is designed, and maybe implemented
- Your dictionary is designed (in Session 103)
- You have identified one or more real-world use cases
  - A feature extension
  - Implemented as an AppleScript
  - Solving an actual end-user problem
- You have access to the sample code

# Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

# For Cocoa or Carbon

- Carbon requires setup, come to lab

Related session

**AppleScript Lab**

**Wednesday 2:00pm**

# What You'll Learn

- Logging
- Element accessors
- Property accessors
- Commands
  - Object-first dispatching
  - Verb-first dispatching
- Automated regression testing
- Testing real-world use cases

# Logging

- Simple

```
#if  0
#define jgcLog1(x)          NSLog(x)
#define jgcLog2(x,y)        NSLog(x,y)
#else
#define jgcLog1(x)
#define jgcLog2(x,y)
#endif
```

- Ubiquitous
  - Log every accessor call
  - Log every event-handler call
  - Log significant processing phenomena

# Element Accessors

## “Getter” functions

- Simple

```
-(NSArray*)rectangles
```

- Accelerated

```
-(id)valueInRectanglesAtIndex:(unsigned)index;
-(id)valueInRectanglesWithName:(NSString*)name;
-(id)valueInRectanglesWithID:(id)uniqueID;
```

# Element Accessors

## "Setter" functions

- Make

```
-(void)insertInRectangles:(id)rectangle;
-(void)insertInRectangles:(id)rectangle atIndex:(unsigned)index;
```

- Delete

```
-(void)removeFromRectanglesAtIndex:(unsigned)index;
```

- Replace

```
-(void)replaceInRectangles:(id)rectangle atIndex:(unsigned)index;
```

# Demo

## Logging and Element Accessors

http://developer.apple.com/wwdc2005/

# Property Accessors

- Read-only
- Read-write
- Leverage Key-Value Coding (KVC)
  - NSDocument
  - Cocoa bindings

# Demo

## Property Accessors

http://developer.apple.com/wwdc2005/

# Commands

- Object-first dispatching

```
rotate rectangle 1 by 90
```

- Verb-first dispatching

```
align every rectangle by vertical center
```

# Commands

- Object-first dispatching
  - Command dictionary entry

```
<command name="rotate" code="Skchrota" description="Rotate
graphic object(s)">
   <cocoa class="NSScriptCommand"/>
   <parameter name="by" code="by  " type="real"
optional="no" description="Degrees counterclockwise by
which to rotate the object(s).">
      <cocoa key="byDegrees"/>
   </parameter>
</command>
```

# Commands

- Object-first dispatching
  - Responds-to dictionary entry

```
<class name="graphic" code="grfc" description="A graphic
object.">
   <cocoa class="SKTGraphic"/>
...
   <responds-to name="rotate">
      <cocoa method="rotate:"/>
   </responds-to>
...
</class>
```

  - Instance method

```
-(id)rotate:(NSScriptCommand*)command;
```

# Demo

## Commands, object-first dispatching

http://developer.apple.com/wwdc2005/

# Commands

- Verb-first dispatching
  - Command dictionary entry

```
<command name="align" code="Skchalin" description="Align
multiple graphic objects">
   <cocoa class="SKTAlignCommand"/>
   <parameter name="by" code="by  " type="edge"
optional="no" description="Edge or center to which to
align the objects.">
      <cocoa key="toEdge"/>
   </parameter>
</command>
```

# Commands

- Verb-first dispatching
  - Subclass NSScriptCommand
  - Override performDefaultImplementation

```
@interface SKTAlignCommand : NSScriptCommand
{
}

-(id)performDefaultImplementation;

@end
```

# Demo

## Commands, verb-first dispatching

http://developer.apple.com/wwdc2005/

# Commands

- Asynchronous
  - Suspend

```
[command suspendExecution]; //  object-first

[self suspendExecution];    //  verb-first
```

  - Resume

```
[[NSScriptCommand currentCommand]
  resumeExecutionWithResult:result];
```

# Test Scripts

- Elements
  - get
    - every
    - by name
    - by id
    - by index

# Test Scripts

- Elements
  - make
  - delete
  - replace

# Test Scripts

- Properties
  - get
    - All properties

```
get properties of rectangle 1
```

    - by name

```
get height of rectangle 1
get width of rectangle 1
```

# Test Scripts

- Properties
  - set
    - get/set/reset
    - edge cases
    - error cases

# Test Scripts

- Commands
  - Parameters
    - Default
    - Specified
  - Edge cases
  - Error cases

# Testing Use Cases

- Real world
- Repeatable

# Review

- Logging
- Element accessors
- Property accessors
- Commands
  - Object-first dispatching
  - Verb-first dispatching
- Automated regression testing
- Testing real-world use cases

# Documentation, Sample Code, and Other Resources

http://developer.apple.com/wwdc2005/

# Related Sessions

| | |
|---|---|
| AppleScript Lab | Wednesday 2:00pm |

# Who to Contact

**Sal Soghoian**
AppleScript and Automator Product Manager
sal@apple.com

**Todd Fernandez**
AppleScript and Automator Engineering Manager
trf@apple.com

**AppleScript Implementors**
Mailing List
applescript-implementors@lists.apple.com

# Q&A

**John Montbriand**
DTS Engineer

http://developer.apple.com/wwdc2005/

# WWDC2005

Worldwide Developers Conference